

SOUTENANCE 1

2020-2021



Team ALPHA



Alexandre Devaux-Riviere

Léo Devin

Phu Hung Dang

Angélo Eap

Table des matières

1	Prototype de logo - ERASED	4
2	Menu	4
2.1	Menu Principal	4
2.1.1	Start solo :	4
2.1.2	Start multi :	4
2.1.3	Settings :	4
2.1.4	Quit Game :	4
2.2	Menu Options	5
2.2.1	Paramètres de son	5
2.2.2	Résolution de la fenêtre	5
2.2.3	Mode plein écran	6
2.2.4	Réinitialisation des données de la partie	6
2.2.5	Quitter le menu option	6
3	Personnage principal	6
3.1	Déplacement	6
3.2	Design du personnage	7
3.3	Points de vie	7
3.3.1	Cœurs	7
3.3.2	Dégâts	8
3.4	Caméra	8
4	Inventaire	8
4.1	Objet ramassable	8
4.1.1	Potions	8
4.1.2	Pièces	8
5	Affichage-HUD	9
5.1	Images par secondes (FPS)	9
5.2	Compteur de pièces	9
5.3	Barre de vie	9
5.4	Fin du jeu (GameOver)	10
5.5	Sélection du niveau	10
6	Level design	11
6.1	Zone d'élimination	11
6.2	Points de sauvegarde	11
6.3	Mort du personnage	11
6.4	Musique et son	11
6.5	Système de sauvegarde de données	12
6.6	Déverrouillage des niveaux	12
6.7	Transitions	12
6.8	Sauvegarde d'objets	12

7	Apparence du terrain (Map)	13
7.1	Les différentes couches	13
7.1.1	Le sol et les murs	13
7.1.2	Les décorations	14
7.1.3	Les détails	14
7.2	Décors interactifs	14
7.2.1	Portes	14
7.2.2	Coffres	15
7.2.3	Échelles	15
8	Réseau	15
8.1	Photon Unity Network	15
8.2	Interface Multijoueur (Lobby)	15
8.3	Salles multijoueur (Rooms)	16
8.4	Instanciation du joueur dans la room	17
8.5	Les ajouts directement en jeu Multijoueur	18
9	Intelligence Artificielle	18
9.1	IA - active	18
9.2	IA - passive	19
9.2.1	PNJ avec une boîte de dialogue	19
9.2.2	PNJ vendeur	19
10	Communication	20
11	Modification des répartitions des tâches	21
12	Futures améliorations	22
12.1	Attaque du personnage principal	22
12.2	Armes	22
12.3	Ajout de niveaux	22
12.4	Changement des portes	22
12.5	Ajout d'ennemis et de boss	22
12.6	Musique et son	22
12.7	Multijoueur	22
12.8	IA	23
13	Avances et retards	23

1 Prototype de logo - ERASED



FIGURE 1 – prototype : ERASED nouveau logo

2 Menu

2.1 Menu Principal

Le menu principal est la première chose que le joueur va voir lors du lancement du jeu, il est donc très important. Bien que rudimentaire pour le moment, il sera amélioré en y ajoutant des effets visuels et sonores par la suite. Le menu principal propose quatre boutons d'interaction (user interface (UI)), les voici :

2.1.1 Start solo :

Ce bouton va rediriger le joueur vers le menu de sélection de niveau qui sera détaillé plus tard dans la présentation, le joueur se lance alors dans une aventure en solitaire.

2.1.2 Start multi :

Comme le fait le bouton précédent, le bouton "Start multi" dirigera le joueur vers le menu d'hébergement *Photon* pour pouvoir rejoindre ou créer une salle en multijoueur, cette interface est décrite plus tard.

2.1.3 Settings :

Le bouton "Settings" comme sa traduction anglaise l'indique, amène le joueur dans le menu des paramètres, comme les paramètres de son et bien d'autres. Ce menu est expliqué en détail dans la section qui suit.

2.1.4 Quit Game :

Ce bouton fermera simplement le jeu et faire revenir l'utilisateur sur son bureau.



FIGURE 2 – Menu Principal

2.2 Menu Options

Afin que le joueur puisse choisir ses options, nous avons implémenté un menu permettant de choisir plusieurs paramètres essentiels au jeu. Ces options sont atteignables et modifiables à l'aide de *Unity* qui facilite l'accès à ces options. Voici les différentes options que nous avons implémentées pour l'instant :

2.2.1 Paramètres de son

Ce paramètre possède deux "sliders" différents. Un slider est une barre où se situe un bouton qui peut glisser d'un point A à un point B, ces deux points représentent le son minimum et le son maximum ; respectivement, de la gauche vers la droite. Le premier slider va modifier le volume de la musique uniquement. Le deuxième lui modifie le volume des effets sonores.

Afin que le son soit réglé correctement, nous travaillons en décibels, le son maximum correspond à 0 décibel, ce qui est déjà un volume élevé. Il peut diminuer jusqu'à -80 décibels, qui est un son inaudible. Lorsque le jeu est lancé, les sons sont configurés sur le volume maximum.

2.2.2 Résolution de la fenêtre

La résolution est la taille de la fenêtre de jeu en nombre de pixels. Nous permettons ici au joueur de choisir la résolution qui lui convient. Le système de résolution va détecter automatiquement toutes les résolutions que l'écran de l'utilisateur peut prendre en charge. Il pourra de ce fait stocker toutes les valeurs possibles dans un tableau. C'est ce tableau défilant que le joueur utilisera pour choisir sa résolution. Lorsque le joueur lance le jeu pour la première fois, la résolution correspondant le mieux à l'écran de celui-ci sera automatiquement choisi, mais elle peut être changée si le joueur n'est pas satisfait.

2.2.3 Mode plein écran

Le mode plein écran, comme son nom l'indique, permet au joueur de choisir si le jeu remplit l'entièreté de son écran ou non. Par convention, le mode plein écran est activé par défaut lorsque le joueur lance le jeu pour la première fois.

2.2.4 Réinitialisation des données de la partie

Nous avons un système de sauvegarde qui mémorise le déverrouillage des niveaux par le joueur. Nous avons donc implémenté un bouton qui permet de réinitialiser les données du jeu à zéro. Cela signifie donc qu'il perdra tout son avancement dans le jeu, ses pièces et son inventaire. Cette option n'est présente que dans les options du menu principal. En effet, le joueur ne peut pas effacer ses données alors qu'il est en pleine partie !

2.2.5 Quitter le menu option

Lorsque le joueur a fini de modifier les paramètres, pour revenir au menu précédent, il y a un bouton en haut à droite de la fenêtre permettant de revenir à l'interface précédente.



FIGURE 3 – Menu Options

3 Personnage principal

3.1 Déplacement

Le déplacement du personnage se fera à l'aide des touches "espace" et "←", "↑", "↓", "→" ou "Q", "Z", "S", "D". Nous avons utilisé une "Capsule Collider 2D" qui est une hitbox en forme de capsule empêchant le joueur de traverser les structures solides tels que les murs et les sols. Le personnage dispose d'un "GroundCheck" constitué d'un Gizmo circulaire directement situé en dessous du personnage et synchronisé avec celui-ci. En effet, le "GroundCheck" vérifie s'il est en contact avec le sol et renvoie un booléen en fonction de cette condition. Le joueur peut donc sauter seulement au contact du sol, évitant ainsi les sauts à l'infini dont nous avons fait l'expérience. Afin que son déplacement semble plus dynamique,

nous avons également augmenté la gravité ainsi que la puissance du saut, car le personnage semblait flotter lorsqu'il sautait. Le déplacement en général fait intervenir des fonctions de déplacements plus douces et réalistes que celles que l'on peut trouver dans la plupart des jeux 2D simulant ainsi l'inertie du personnage lié à son état initial.

3.2 Design du personnage

Le personnage n'est pas notre création, mais nous l'avons téléchargé car son visuel nous plaisait, et l'avons modifié pour qu'il corresponde au thème choisi. Le personnage est donc assez pixelisé, muni d'une épée fluorescente et d'une cape noire. Il existe aussi une version du personnage avec d'autres armes (prochainement avec un arc et une lanterne magique). D'autres assets présents dans le jeu ont également été téléchargés et utilisés pour les différentes entités. Le personnage principal possède plusieurs animations, une immobile lorsqu'il ne se déplace pas et une autre lorsqu'il court. Nous possédons également d'autres animations comme par exemple une animation d'attaque ou encore d'autres capacités telles que les dashes, mais l'implémentation de ces derniers n'est pas disponible pour l'instant.

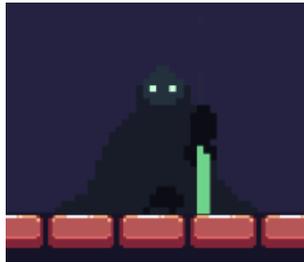


FIGURE 4 – Personnage Principal

3.3 Points de vie

Le joueur possède un nombre de points de vie, lorsque ses points de vie atteignent zéro, celui-ci meurt. Cependant le joueur peut récupérer ses points de vie, notamment par l'usage des potions ou en ramassant des coeurs présents dans chaque niveau. De plus, lorsque le joueur termine un niveau, il récupère tout ses points de vie.

3.3.1 Coeurs

Nous avons implémenté un système permettant au joueur de récupérer ses points de vie lorsqu'il prend un coeur. Ces coeurs se trouvent un peu partout dans le niveau afin d'aider le joueur.

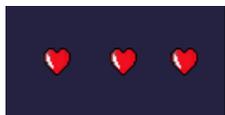


FIGURE 5 – Texture des coeurs

3.3.2 Dégâts

Le joueur perd des points de vie lorsque celui-ci se fait attaquer par un ennemi. Nous avons actuellement peu de situations dans lesquelles le joueur prend des dégâts, mais des pièges et d'autres ennemis seront implémentés dans le futur.

3.4 Caméra

Nous avons procédé à la création de deux caméras synchronisées avec la position du personnage. Tout d'abord une première caméra qui reproduit les déplacements du personnage en différé d'environ 0,3 secondes donnant un effet de transition lisse à chaque déplacement du personnage. Puis, nous avons implémenté une caméra beaucoup plus dynamique qui est synchronisée cette fois avec la vitesse et la direction du personnage depuis l'extension "Cinémachine" directement disponible depuis *Unity*. Cependant, il est important de trouver un bon équilibre dans la construction de cette caméra car elle peut très vite fatiguer le joueur.

4 Inventaire

Des objets pourront être consommés à partir d'un inventaire. De plus, le joueur pourra naviguer entre les objets qu'il possède en cliquant sur les flèches de l'inventaire. L'objet présent dans la case du milieu pourra être consommé. L'inventaire se situe en bas à droite de l'écran. Pour l'instant, les seuls objets consommables sont des potions, mais d'autres objets feront leur apparition très prochainement.

4.1 Objet ramassable

4.1.1 Potions

Le joueur pourra ramasser des objets, pour l'instant il n'y a qu'une potion de soin et de vitesse, qui respectivement soigne le joueur d'un montant de points de vie que l'on peut modifier, et augmente sa vitesse de déplacement d'un certain montant durant une certaine durée. Nous allons prochainement créer d'autres potions, comme une potion de saut.

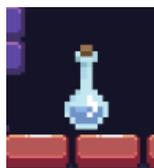


FIGURE 6 – Texture des potions

4.1.2 Pièces

Nous avons ajouté un système de pièces basé sur le gain d'une pièce lorsque le joueur passe sur cette dernière. Ses pièces sont sauvegardés seulement lorsque le joueur change de niveau, donc si le joueur meurt durant l'exploration du niveau, les pièces récupérées dans celui-ci ne seront pas comptées lorsqu'il réapparaîtra.



FIGURE 7 – Texture des pièces

5 Affichage-HUD

5.1 Images par secondes (FPS)

Nous avons ajouté un compteur et un limiteur de FPS (60 FPS), dû à un bug de ralentissement du jeu sans ce dernier. Lorsque les FPS sont en dessous de 20 le compteur devient rouge, s'il est en dessous de 30, il sera orange et pour toute autre valeur il sera vert.



FIGURE 8 – Compteur de FPS

5.2 Compteur de pièces

L'affichage du total des pièces du joueur est un compteur, il se trouve en haut à droite de l'écran. Le compteur augmente de un lorsque le joueur ramasse une pièce au sol et de dix lorsqu'il ouvre un coffre.



FIGURE 9 – Compteur de pièces

5.3 Barre de vie

Pour avoir un visuel de la vie du joueur, nous avons ajouté une barre de vie en haut à gauche de l'écran. Lorsque le joueur prend des dégâts celle-ci diminue et inversement lorsqu'il regagne de la vie. La barre est verte lorsque le personnage a beaucoup de points de vie, et vire au rouge lorsqu'il passe un certain pallier.



FIGURE 10 – Barre de vie



FIGURE 11 – Barre de vie lorsque le joueur prend des dégats

5.4 Fin du jeu (GameOver)

Nous avons ajouté une interface de fin de jeu, lorsque le joueur meurt, un menu s'affiche et lui demande s'il veut rejouer, et à l'instar du menu pause, d'aller dans le menu principal ou quitter la partie.



FIGURE 12 – Ecran mort du joueur

5.5 Sélection du niveau

Nous avons créé une interface permettant au joueur de sélectionner un niveau après avoir choisi le mode solo dans le menu principal. Seul le premier niveau est sélectionnable au début. Les autres niveaux seront par la suite débloqués lorsque le joueur réussira à finir le niveau qui le précède.



FIGURE 13 – Ecran sélection de niveau

6 Level design

6.1 Zone d'élimination

La zone d'élimination est une zone qui fera réapparaître le joueur s'il la touche, il est utilisé pour simuler du vide ou une zone dangereuse comme par exemple des piques.

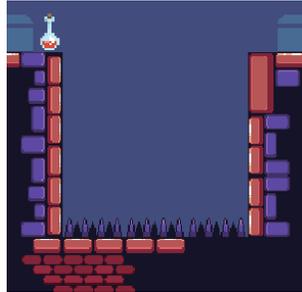


FIGURE 14 – Zone d'élimination

6.2 Points de sauvegarde

Les points de sauvegarde ou checkpoints permettent au joueur de revenir à un certain point de sauvegarde lorsqu'il rentre dans une zone d'élimination et non au tout début du niveau.



FIGURE 15 – Texture du checkpoint

6.3 Mort du personnage

Nous avons ajouté une animations de mort (personnage qui part en fumée) lorsque le personnage meurt. Le joueur "meurt" lorsqu'il perd tout ses points de vie (ses points de vie sont inférieure ou égale à 0), le menu de fin de partie s'affiche avec les options "retry", "main menu" et "quit".

6.4 Musique et son

Nous avons pour l'instant deux musiques de jeu qui vont se jouer en boucle l'une après l'autre, ainsi que des effets sonores qui se déclenchent quand une certaine action est réalisée (ouverture de coffre, récupération d'un coeur, d'une pièce). Ces musiques n'ont pas été créées mais téléchargés à partir de sites proposant des assets.

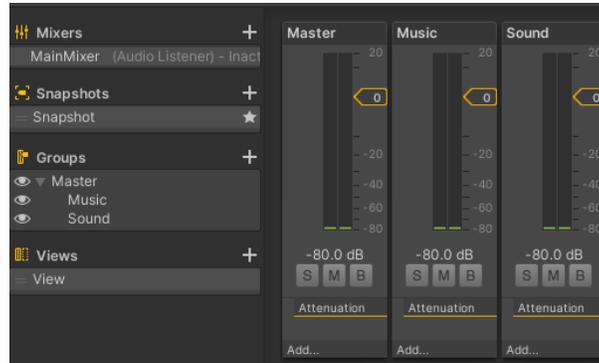


FIGURE 16 – Musique et son

6.5 Système de sauvegarde de données

Le système de sauvegarde permet au joueur de ne pas perdre toutes ses données telles que les pièces, ses niveaux débloqués ou ses paramètres de jeu lorsqu'il le quitte.

6.6 Déverrouillage des niveaux

De plus, nous avons implémenté un système de déverrouillage de niveau. En effet le joueur n'a accès qu'au premier niveau au début du jeu comme dit précédemment, et débloque les suivants une fois le niveau en question terminé. Le niveau suivant pourra alors être sélectionné dans le menu. Par ailleurs, il pourra aussi rejouer aux anciens niveaux.

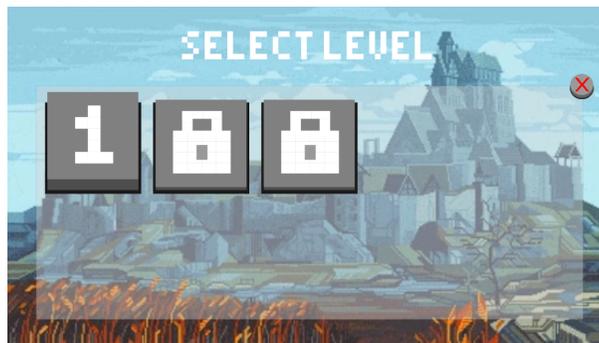


FIGURE 17 – Niveaux bloqués

6.7 Transitions

La transition de la caméra entre deux scènes étant brusque, nous avons décidé de créer une transition, un fondu au noir, qui est un dégradé de noir permettant un meilleur visuel à chaque fois que l'on rentre ou sort d'une scène.

6.8 Sauvegarde d'objets

L'ajout d'un système de sauvegarde de l'inventaire d'objet permet au joueur de ne pas perdre ses objets entre chaque scène.

7 Apparence du terrain (Map)

Unity nous permet de modifier notre terrain à l'aide de "Tiles map", ce sont des plans qui regroupent des petits carrés (les tiles), chaque tile aura sa propre texture. Nous avons trouvé un asset (texture pour nos tiles et d'autres objets décoratifs) que nous utilisons majoritairement sur nos terrains. En assemblant toutes ces textures, nous pouvons donc former une carte (plus communément appelée map). Pour pouvoir superposer des textures, être mieux organisé et mieux travailler, nous regroupons certaines tiles en couches (ou "layers") qui auront chacune un index de profondeur différentes. Plus l'index sera élevé, plus la couche sera présente au premier plan.

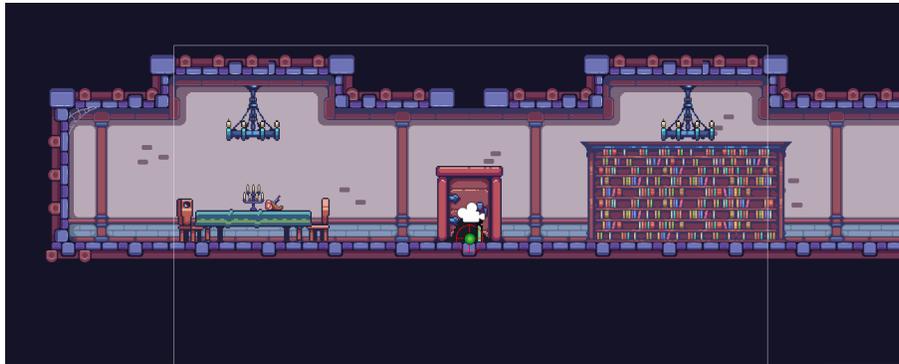


FIGURE 18 – Terrain en construction

7.1 Les différentes couches

7.1.1 Le sol et les murs

Cette layer est dédiée à tous les éléments avec lesquels le joueur rentre en collision, c'est à dire qu'il ne pourra pas passer à travers. Dans cette layer nous comptons donc le sol et les murs, mais aussi certains éléments du terrain servant à obstruer le passage du joueur. Pour cela, il faut ajouter un "component", un élément propice à chaque objet et entité de la scène. Ce component est un collider, et comme dit précédemment, il va empêcher les entités ayant une boîte de collision (comme le joueur par exemple) de passer à travers. Le component va faire en sorte que la boîte de collision épouse parfaitement les formes des tiles présentes sur la layer, c'est à dire que si la tile est un escalier, la boîte ne sera pas un cube mais bien un triangle rectangle prenant la forme de l'escalier en question. Cette layer va avoir un index important par rapport aux autres layers, parce qu'elle est située au premier plan de la scène.

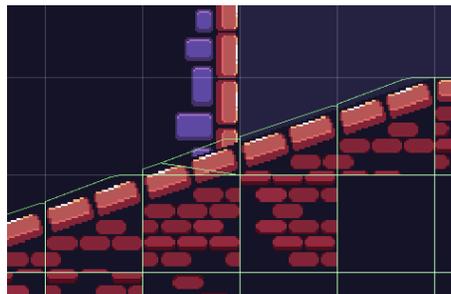


FIGURE 19 – La layer sol et murs

7.1.2 Les décorations

La layer comprenant les décorations va avoir un index inférieur à celui du sol et des murs car ce sont des éléments présents en arrière plan. On peut y trouver par exemple le fond, les objets de décorations comme des tables, des fenêtrés ou bien même des bibliothèques, ces derniers servent de décor car le joueur peut passer à travers. Cette layer va être celle qui comprend le plus d'éléments lorsque tous les détails seront implémentés

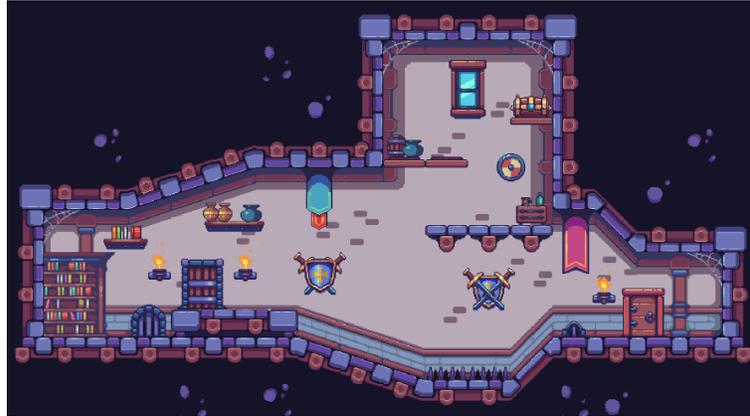


FIGURE 20 – Les décorations

7.1.3 Les détails

Cette layer comme son nom l'indique est présente pour combler les petits détails pouvant manquer sur les layers, comme les effets de briques sortant des murs, ou encore les roches présentes dans les sols.

7.2 Décors interactifs

Les décors interactifs sont des éléments du décor avec lesquels le joueur peut interagir. Lorsque le joueur s'approche d'un de ces objets un message apparaîtra sur l'écran.

7.2.1 Portes

Les portes permettent aux joueurs d'accéder au prochain niveau ou à différentes salles, elles peuvent être utilisées en passant près de ces dernières.

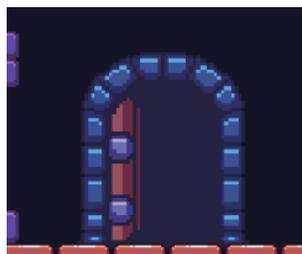


FIGURE 21 – Porte permettant le changement de scène

7.2.2 Coffres

Les coffres permettent aux joueurs de gagner une certaine somme de pièces en l'ouvrant, pour cela le joueur doit s'approcher du coffre et appuyer sur la touche "E".



FIGURE 22 – Animation du coffre

7.2.3 Échelles

L'ajout des échelles permet au joueur de monter sur des plate-formes en hauteur non atteignables avec un simple saut. Pour utiliser l'échelle, il vous sera dit d'appuyer sur W ou Z selon la langue de votre clavier, le personnage pourra alors se déplacer verticalement jusqu'à qu'il atteigne la fin de l'échelle, ou qu'il redescende au sol.

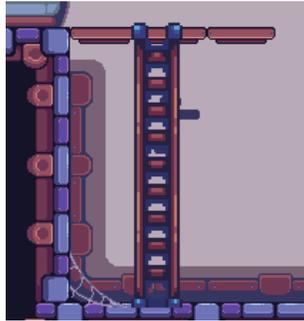


FIGURE 23 – Echelle

8 Réseau

8.1 Photon Unity Network

Pour gérer la partie multijoueur de notre jeu, nous avons choisi d'utiliser l'asset "Photon Unity Network" (PUN) qui permet avec la version gratuite d'avoir jusqu'à 20 joueurs présents en même temps en multijoueur sur le jeu. Le principe du fonctionnement choisi est le suivant : une fois l'option multijoueur sélectionnée, les joueurs se connectent dans un premier temps au serveur *Photon* puis une interface de sélection (lobby) leur est proposée permettant ensuite d'accéder aux options de création et de connexion des Rooms, c'est-à-dire des sessions multijoueur.

8.2 Interface Multijoueur (Lobby)

Dans le Lobby, il est possible de :

- Créer une room
- Lister des rooms déjà existantes
- Joindre une room
- Quitter le lobby et revenir au menu principal

Afin d'obtenir des informations sur l'état de la connexion avec le serveur *Photon*, nous avons créé le script "worker-scene.cs" contenant des fonctions avec diverses vérifications de connexion rendus sous forme d'informations au joueur. Ces informations sont rendues visibles grâce à la liaison entre ces fonctions et un simple canvas avec du texte qui s'affiche avant que l'interface Multijoueur n'apparaissent. Si la connexion ne peut être établie correctement, un message d'erreur apparaît et l'interface multijoueur n'est pas présentée au joueur.

8.3 Salles multijoueur (Rooms)

Dans une room il est possible de :

- Lister les connexions et déconnexion des joueurs
- Instancier des objets
- Autres actions situationnelles

Pouvoir joindre et créer une room a nécessité des ajouts dans le script "worker-scene.cs". Celui-ci possède désormais des fonctions qui permettent de générer des boutons associés à des actions ainsi que des zones de texte où le joueur peut écrire. En effet, il est possible de rejoindre une room spécifique en écrivant exactement son nom dans le champ adéquat (via l'inputfield de la room) avant de rejoindre, d'en joindre une de façon aléatoire ainsi que d'en créer une si la room n'existe pas déjà, sachant que par défaut elle est nommée "My Room".

De plus, il est possible de choisir un pseudo (via l'inputfield du joueur) qui sera conservé pendant toute la durée de la partie multijoueur. Pour un souci de confort, les pseudos suivent par défaut une typographie de ce genre : "Guest" suivi d'un nombre aléatoire entre 100 et 9999 compris et c'est ainsi aux joueurs de le changer s'ils le désirent sachant que le pseudo fera toujours au minimum un caractère.

En outre, des informations sont aussi données quant au trafic sur le jeu. C'est à dire que le nombre de joueurs actuellement présents en multijoueur ainsi que le nombre de rooms sera toujours affiché avec en plus une liste des rooms déjà actives que l'on peut faire défiler afin de rejoindre celle que l'on souhaite. Pour finir, l'esthétique de l'ensemble de l'interface a été liée à un "GUI Skin" permettant de modifier les typographies ainsi que les graphismes des parties de l'interface directement depuis l'extérieur des scripts, ce qui est très pratique pour nous. Il est aussi important de spécifier que la plupart des cas d'erreurs sont pris en compte, gérés et affichés au joueur, permettant d'assurer l'intégrité de notre jeu.



FIGURE 24 – Interface Multijoueur

8.4 Instanciation du joueur dans la room

Pour que les joueurs puissent se voir dans la room, il faut tout d'abord instancier le joueur. Pour ce faire, la première étape consiste à ajouter le composant "photonview" à l'objet correspondant au joueur en renseignant l'élément que *Photon* doit observer donc ici le personnage principal du jeu.

Par la suite, il a fallu instancier notre objet lors du lancement de la scène en se rendant dans le script "GameManagerMulti.cs" pour y ajouter le GameObject correspondant au joueur afin de pouvoir l'instancier lors du lancement du jeu.

Cependant, cette manipulation a nécessité que l'objet de notre joueur et donc sa prefab soit placée dans le dossier Ressources de PUN. Maintenant que nos joueurs sont instanciés, il faut les différencier (sinon chaque joueur déplace tous les personnages). Donc un script "Photonview" initialisé au start a été ajouté au script "PlayerMulti.cs". De cette manière, chaque joueur est assigné à un unique personnage qu'il est le seul à contrôler, et cela va de pair avec la caméra qui, une fois intégrée à l'objet du personnage, est unique au personnage et donc au joueur évitant tout problème de partage de caméra entre joueurs et caméras fixes.



FIGURE 25 – Exemple situation 1



FIGURE 26 – Exemple situation 2

8.5 Les ajouts directement en jeu Multijoueur

En jeu multijoueur il est possible d'avoir :

- Message de prévention
- Le ping unique à chacun des joueurs
- Le pseudo affiché en haut de chacun des joueurs
- Barre de vie

Nous avons tout d'abord implémenté en multijoueur les mouvements du personnages synchronisés avec le serveur ainsi que ses changements de sens en fonction des déplacement du dit joueur. Nous avons implémenté un message de "prévention" afin de réguler un minimum tout les comportements allant à l'encontre d'une expérience plaisante sur le jeu sachant que celui-ci vise, au premier abord, à détendre le joueur. Passer ce message nous amène directement dans le niveau multijoueur.

Pour la partie concernant les pseudo, nous les avons synchronisés pour que chaque joueur puisse voir le pseudo des autres. En effet, une fois le pseudo rentré dans l'interface multijoueur, il est affiché dans un UI de texte toujours au-dessus du joueur concerné (car lié à la prefab du joueur). Nous avons aussi ajouté une option permettant d'afficher notre pseudo en blanc ainsi que ceux des autres en cyan afin de se repérer plus facilement.

De plus, nous avons ajouté un champ de texte qui intervient pour renseigner la connexion ou la déconnexion d'un joueur sans jamais superposer le texte. En effet, nous avons créé un "Grid Layout" à gauche du point de vu de tous les joueurs dans lequel les UI de textes apparaissent et se décalent d'une distance prédéfinie tant que le texte renseignant une ancienne connexion ou déconnexion n'ai pas disparu.

Nous avons aussi ajouté le ping unique de chaque joueur encore une fois dans un UI de texte en haut à gauche. Les barres de vies ont aussi été ajoutés au dessus de chacun des joueurs, cependant elles ne sont pas encore fonctionnelles et devraient l'être d'ici la prochaine soutenance.

Bien évidemment, une option est disponible pour quitter le mode multijoueur si la touche "echap" est actionnée.

9 Intelligence Artificielle

9.1 IA - active

L'IA est actuellement capable de :

- Suivre le joueur placé en paramètre quand celui-ci pénètre dans son périmètre d'action
- Poursuivre le joueur jusqu'à ce que celui-ci sorte du-dit périmètre
- Revenir à son périmètre lorsque celui-ci est poussé du-dit périmètre
- Suivre un chemin prédéfini

Dans l'optique de débiter les tests concernant les mécaniques de jeu, nous avons commencé par implémenter une IA minimaliste dont le but est de se rendre d'un point à un autre en administrant des dégâts au contact avec notre personnage principal et disparaissant si ce contact a lieu sur la tête de l'IA.

Dans un second temps, nous avons créé une IA qui se contente de vérifier sa distance avec un joueur placé en paramètre de son script et de le prendre en chasse si celui-ci rentre dans son périmètre. Cela a permis de tester certaines fonctionnalités comme sa perception de l'univers dans lequel elle évolue avec notamment sa capacité à éviter les obstacles tout en évaluant le chemin le plus court pour rattraper la cible.

Lors de la création de cette IA, un problème s'est posé au niveau de sa conception même. En effet, dans un premier temps nous avons essayé d'implémenter une IA à l'aide d'un asset de "path finding" gratuit mais il s'est avéré que malgré le bon fonctionnement de celle-ci, sa physique la rendait hors d'usage. Il était impossible de l'arrêter ou même de la freiner en la poussant, laissant place à un joueur qui pouvait être coincé entre l'IA et un obstacle sans possibilité de mouvement, ce qui est problématique pour le gameplay ainsi que la cohérence dans la physique du jeu. C'est pour cette raison que nous avons nous même construit une IA qui remplit le rôle de traquer les joueurs dans une zone donnée sans pour autant être dénuée de physique.

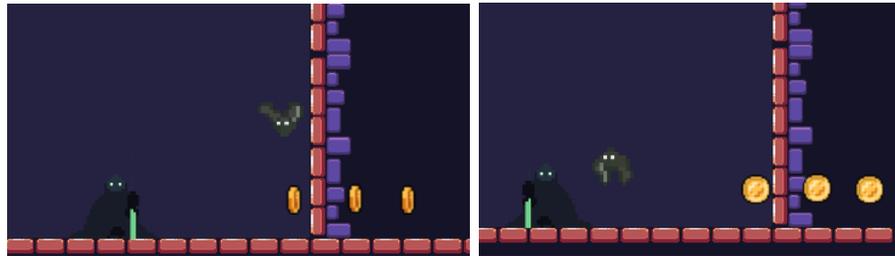


FIGURE 27 – IA : Avant / Après

9.2 IA - passive

Un PNJ est un Personnage Non Joueur. Plusieurs types de PNJ seront présents dans le jeu, pour l'instant nous avons un PNJ qui parle au joueur et un autre qui vend des objets. Tout PNJ possède un nom et un design qui les différencient les uns des autres.

9.2.1 PNJ avec une boîte de dialogue

Lorsque le joueur s'approche du PNJ et qu'il appuie sur la touche "E", une boîte de dialogue apparaît sur l'écran permettant au PNJ de communiquer avec le joueur. Ce dialogue peut à tout moment être modifié.

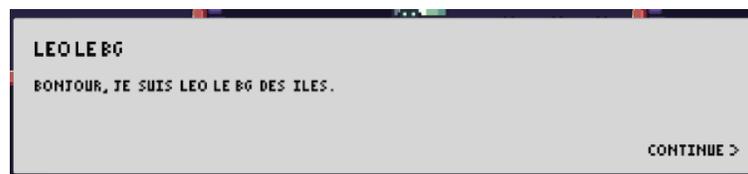


FIGURE 28 – Interaction PNJ dialogue

9.2.2 PNJ vendeur

Nous avons implémenté un PNJ qui aura la capacité de vendre des potions au joueur. Lorsque le joueur s'approche du vendeur et qu'il interagit avec, en appuyant sur la touche "E", des objets apparaîtront. Ces objets vont coûter un certain montant de pièces que le joueur aura stocké au fil de la partie. Les objets ainsi que leur prix peuvent être modifiés.



FIGURE 29 – Interaction PNJ vendeur

10 Communication

Nous avons utilisé *WIX*, un créateur de site en ligne pour faire un site pour notre jeu. Sur le site, vous pouvez y trouver des informations sur le jeu tels que des captures d'écran du jeu, l'avancement du jeu, des informations sur l'ensemble de l'équipe, un lien de téléchargement du cahier des charges ou encore du jeu. Cependant, si nous possédons un peu de temps après avoir terminé la création du jeu, nous essaierons d'en créer un nous-même. De plus, nous avons créé un *Twitter* au nom de *projet_erased* qui nous sert à communiquer sur l'avancement du projet et des problèmes rencontrés lors de sa création.

Liens :

Site Web : <https://projecterased.wixsite.com/erased>

Twitter : https://twitter.com/project_erased

11 Modification des répartitions des tâches

	Alexandre	Léo	Phu Hung	Angélo
Menu				
Ecran d'accueil		X	*	
Accès Modes (Multijoueur, solo, ...)		X		*
Accès option In-Game		X	*	
Mécaniques de jeu				
Caméra	X		*	
Implémentation maps (Spawns, ...)	*			X
Animations (entités, armes, ...)	*		X	
Graphisme				
Maps solo / multi		X		*
Scénario			X	*
Personnages (Héros, entités, ...)	X			*
Environnement (Ombres, ...)	X	*		
Son - Atmosphère				
Bruitages		X	*	
Musiques		*	X	
Intelligence Artificielle				
Entités	X			*
Boss	X			*
Evènements liés à la map		*		X
Système Réseau				
LAN	-	-	-	-
Serveur	X		*	
Mode multijoueur	X		*	
Communication				
Site de présentation	X	X	X	X
Rendus Overleaf	X	X	X	X

Legende :

X = Leader

* = Contributeur

12 Futures améliorations

12.1 Attaque du personnage principal

Notre personnage bouge et saute, certes, mais il lui faudrait une fonction d'attaque qui lui permettra d'éliminer les ennemis, ce qui rajoutera plus d'action au jeu.

12.2 Armes

Pour l'instant, notre personnage possède une épée, et ne peut que se déplacer avec. Nous voulons implémenter un arc afin qu'il puisse attaquer à distance, et peut-être d'autres armes si possible.

12.3 Ajout de niveaux

Nous avons trois niveaux pour faire des tests, mais il est nécessaire dans le futur d'en avoir des vrais, avec plus de décor et d'ennemis. Nous souhaitons par exemple créer un niveau qui se passera à l'extérieur, il faudra donc trouver d'autres assets.

12.4 Changement des portes

Nous voulons changer l'interaction avec les portes. En effet, pour l'instant il suffit d'être en contact avec la porte pour l'utiliser. Cependant, nous voulons que le joueur puisse décider d'utiliser ou non la porte, en appuyant sur une touche.

12.5 Ajout d'ennemis et de boss

Les ennemis présents dans le jeu actuellement sont une chauve-souris et un serpent. Nous souhaitons implémenter plus d'ennemis, comme des squelettes, des slimes ou encore un minotaure qui rendront le jeu plus intéressant. Il nous manque aussi de la difficulté, qui sera compensée par l'apparition de "Boss", des monstres plus forts et plus résistants.

12.6 Musique et son

Notre musique pour le moment n'est pas définitive, mais nous essaierons de trouver d'autres musiques plus adaptées à notre thème et qui nous plaise. Les effets sonores nous suffisent pour le moment, mais peuvent toujours être améliorés.

12.7 Multijoueur

De nombreux ajouts vont être considérés et arriveront au fil du temps avec notamment la transcription de la partie solo en multijoueur ainsi que de nouvelles fonctionnalités telles qu'un "chat" d'équipe et peut-être une possibilité de partage d'objets. Par ailleurs, Photon View c'est bien, mais pas suffisamment. En effet, les mouvements des personnages sont saccadés et mal synchronisés. Donc, les étapes restantes à faire porteront d'autre part sur :

- La synchronisation des animations (PhotonAnimatorView)
- La synchronisation de la physique

12.8 IA

Nos objectifs à accomplir à terme de la seconde période sont dans un premier temps d'adapter l'IA au mode multijoueur et de pouvoir commencer les tests sur ce mode et dans un second temps d'améliorer la gestion de son environnement, son "intelligence" et ses possibilités en jeu : déplacements plus complexes, et diversification des interactions avec le niveau et le joueur.

13 Avances et retards

Tâche \ Soutenance	Prévision Soutenance 1	Avancement actuel	Prévision Soutenance 2
Menu	30	70	90
Mécaniques de jeu	30	45	70
Graphisme	50	45	75
Son - Atmosphère	20	50	80
Intelligence Artificielle	20	30	60
Système Réseau	50	50	65
Communication	40	35	60

TABLE 1 – Tableau des soutenances (en %)